

**Base de Dados**



Luís Ressonha – 35003

Rafael Sena - 35025

Introdução

Este trabalho foi criado no âmbito da disciplina de base de dados, tem como objetivo o desenvolvimento de uma base de dados para guardar a informação sobre uma companhia de táxis.

O projeto encontra-se dividido em duas partes. Um relatório com as respostas do enunciado em instruções SQL e álgebra relacional, e dois ficheiros: um com a base de dados e outro com as respostas às perguntas.

Na resolução dos exercícios recorremos aos slides que a professora disponibilizou e alguns vídeos tutoriais com o básico do software.

Exercícios

1)

Chaves primárias:

* relação motorista = Nbi
* relação telefone = Nbi, Telefone
* relação modelo = Modelo
* relação táxi = Matrícula
* relação serviço = Matrícula, Data do Inicio
* relação turno = Nbi, Data do Inicio
* relação cliente = Nif
* relação pedido = Nif, Data Pedido

Chaves candidatas:

* relação motorista = Nbi ou NCartaCondução
* relação telefone = Telefone, Nbi
* relação modelo = Modelo
* relação taxi = Matrícula
* relação serviço = Matrícula, Data do Inicio ou Matrícula, Data do Fim
* relação turno = Nbi, Data do Inicio ou Nbi, Data do Fim
* relação cliente = Nif
* relação pedido = Nif, Data do Pedido

Chave estrageiras:

* relação motorista = {}
* relação telefone = relação motorista (Nbi)
* relação modelo = {}
* relação taxi = {}
* relação serviço = relação taxi (Matricula)
* relação turno = relação motorista (Nbi)
* relação cliente = {}
* relação pedido = relação cliente (Nif)

2)

Create table motorista (

Nome varchar(20),

NCartaCond char(15),

DataNasc timestamp,

Nbi INTEGER,

primary key(Nbi)

);

Create table telefone (

Nbi INTEGER,

Telefone INTEGER,

primary key(Nbi,Telefone),

foreign key(Nbi) references motorista(Nbi)

);

Create table modelo (

Marca varchar(30),

Modelo varchar(30),

NLugares INTEGER,

Consumo char(15),

primary key(Modelo)

);

Create table taxi (

Modelo varchar(30),

Ano INTEGER,

Kms char(10),

Matricula varchar(8),

primary key(Matricula)

);

Create table servico (

DataInicio timestamp,

DataFim timestamp,

Kms char(10),

Valor char(5),

Matricula varchar(8),

CoordGPSInic varchar(15),

CoordGPSfim varchar(15),

primary key(Matricula,DataInicio),

foreign key(Matricula) references taxi(Matricula) );

Create table turno (

DataInicio timestamp,

DataFim timestamp,

KmInicio char(10),

KmFim char(10),

Matricula varchar(8),

Nbi INTEGER,

primary key(Nbi,DataInicio),

foreign key(Nbi) references motorista(Nbi)

);

Create table cliente (

Nome varchar(50),

Morada varchar(50),

CodigoPostal varchar(30),

Nif INTEGER,

primary key(Nif)

);

Create table pedido (

Nif INTEGER,

MoradaInicio varchar(50),

CodigoPostalInicio varchar(30),

DataPedido timestamp,

Matricula varchar(8),

DataInicio timestamp,

primary key(Nif,DataPedido),

foreign key(Nif) references cliente(Nif));

3)

i

INSERT INTO modelo VALUES ('Renault', 'Renault Espace', 7, '7L/100Kms');

INSERT INTO taxi VALUES ('Renault Espace', 2015, 123098, '22-AA-22');

ii

INSERT INTO modelo VALUES ('Mercedes', 'MERCEDES CLK', 7, '9L/100Kms');

INSERT INTO taxi VALUES ('MERCEDES CLK', 2014, 234554, '21-AA-22');

iii

INSERT INTO modelo VALUES ('Honda', 'Honda Civic', 5, '5L/100Kms');

INSERT INTO taxi VALUES ('Honda Civic', 2012, 89764, '20-AA-22');

iv

INSERT INTO modelo VALUES ('Mercedes', 'mercedes-benz classe S', 5, '6.5L/100Kms');

INSERT INTO taxi VALUES ('mercedes-benz classe S', 2015, 79744, '19-AA-22');

B)

i

INSERT INTO motorista VALUES ('Manuel Duarte', 'L-123', '14/1/76', '1234');

INSERT INTO telefone VALUES ('1234', '266262626');

INSERT INTO telefone VALUES ('1234', '939393939');

ii

INSERT INTO motorista VALUES ('Fernando Nobre', 'L-124', '14/1/77', '1235');

INSERT INTO telefone VALUES ('1235', '266262627');

INSERT INTO telefone VALUES ('1235', '939393940');

iii

INSERT INTO motorista VALUES ('Anibal Silva', 'L-125', '14/1/78', '1236');

INSERT INTO telefone VALUES ('1236', '266262628');

INSERT INTO telefone VALUES ('1236', '939393941');

iv

INSERT INTO motorista VALUES ('Francisco Lopes', 'L-126', '14/1/79', '1237');

INSERT INTO telefone VALUES ('1237', '266262629');

C)

i

INSERT INTO cliente VALUES ('José Silva', 'Rua António Silva 23', '7100-434 Évora', 600700800900);

ii

INSERT INTO cliente VALUES ('Francisco Passos', 'Rua Manuel Passos 12', '7000-131 Évora', 600700800901);

iii

INSERT INTO cliente VALUES ('Pedro Sousa', 'Rua Joaquim Sousa 21', '7500-313 Évora', 600700800902);

D)

insert into pedido ('600700800900') ('Rua Silva Pais 33') ('7120-212 Évora') (2/1/2016 às 9:00) ('19-AA-22') (2/1/2016 às 8:43)

E)

i

INSERT INTO turno VALUES ('2/1/2016 8:00', '2/1/2016 17:00', 79744, 79944, '19-AA-22', 1234);

ii

INSERT INTO turno VALUES ('2/1/2016 8:00', '2/1/2016 17:00', 89764, 89964, '20-AA-22', 1235);

iii

INSERT INTO turno VALUES ('3/1/2016 8:00', '3/1/2016 17:00', 234554, 234954, '21-AA-22', 1236);

iv

INSERT INTO turno VALUES ('3/1/2016 8:00', '3/1/2016 17:00', 123098, 123498, '22-AA-22', 1237);

4)

ii.

INSERT INTO servico VALUES ('2/1/2016 8:12', '2/1/2016 8:32', 12, 5.25, '19-AA-22', 0.75, 0.76);

iii.

INSERT INTO servico VALUES ('2/1/2016 8:43', '2/1/2016 8:52', 7, 3.25, '19-AA-22', 0.77, 0.79 );

iv.

INSERT INTO servico VALUES ('2/1/2016 8:53', '2/1/2016 9:59', 98, 53.25, '19-AA-22', 0.80, 0.91);

v.

INSERT INTO servico VALUES ('2/1/2016 10:13', '2/1/2016 10:29', 18, 19.25, '19-AA-22', 1.02, 1.12);

vi.

INSERT INTO servico VALUES ('2/1/2016 11:10', '2/1/2016 11:39', 23, 22.25, '19-AA-22', 1.15, 1.25);

vii.

INSERT INTO servico VALUES ('2/1/2016 12:00', '2/1/2016 13:39', 21, 42.25, '19-AA-22', 1.26, 1.29);

viii.

INSERT INTO servico VALUES ('2/1/2016 15:20', '2/1/2016 15:39', 9, 12.25, '19-AA-22', 1.30, 1.32);

5)

(a) -- Matriculas dos taxis Mercedes --

-- Expressão em SQL

select Matricula

from taxi natural inner join modelo

where modelo.Marca = 'Mercedes';

-- Expressão em Álgebra Relacional

Π taxi.Matricula (σ modelo.Marca = 'Mercedes' (taxi |x| modelo))

(b) -- Motoristas que já fizeram um turno num taxi Mercedes --

-- Expressão em SQL

select distinct Nome

from motorista natural inner join turno natural inner join taxi natural inner join modelo

where modelo.Marca = 'Mercedes';

-- Expressão em Álgebra Relacional

Π motorista.nome)(σ(Marca = "Mercedes")(motorista |x| turno |x| taxi |x| modelo))

(d) -- Taxis que já foram conduzidos pelo ’Anibal Silva’ --

-- Expressão em SQL

Select distinct turno.matricula

from turno natural inner join motorista

where motorista.nome = 'Anibal Silva';

-- Expressão em Álgebra Relacional

Π(motorista.telefone)(σ(motorista.Nome = "Anibal Silva" and turno.Nbi = motorista.Nbi and turno.Matricula = taxi.Matricula = serviço.Matricula)(motorista |x| turno |x| taxi |x| serviço))

(e)

-- Expressão em SQL

select motorista.Nome

from motorista

except

select distinct motorista.nome

from motorista, pedido, cliente ,servico, turno

where pedido.Nif = cliente.Nif and cliente.Nome = 'José Silva' and pedido.matricula = servico.matricula and servico.matricula = turno.matricula and turno.nbi = motorista.nbi

-- Expressão em Álgebra Relacional

Π(motorista.nome)(σ(Nome)(motorista))

/

Π(motorista.nome)(σ(pedido.Nif = cliente.Nif and cliente.Nome = "José Silva" and pedido.matricula = serviço.matricula)(motorista |x| turno |x| pedido |x| serviço |x| cliente))

(f)

-- Expressão em SQL

(select distinct Nome

from motorista)

except-

(select distinct Nome

from motorista inner join taxi natural inner join modelo

where taxi.modelo = modelo.Modelo and modelo.Marca = 'Mercedes')

-- Expressão em Álgebra Relacional

Π(motorista.nome)(σ(Nome)(motorista))

/

Π(motorista.nome)(σ(taxi.modelo = modelo.Modelo and modelo.Marca = "Mercedes")(motorista |x| taxi |x|modelo))

(h)

-- Expressão em SQL

select motorista.nome, count(servico)

from motorista natural inner join turno, servico

where turno.datainicio <= servico.datainicio and

turno.datafim>= servico.datainicio and

turno.matricula = servico.matricula

group by motorista.nome;

-- Expressão em Álgebra Relacional

Π(motorista.nome)(g count(serviço)(σ(serviço.DataInicio = turno.DataInicio and serviço.DataFim = turno.DataFim and serviço.Matricula = turno.Matricula)(turno |x| serviço |x| motorista)))